PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} **:**
G06F

A2

(11) International Publication Number: WO 99/66384

(43) International Publication Date: 23 December 1999 (23.12.99)

(21) International Application Number: PCT/US99/13701

(22) International Filing Date: 16 June 1999 (16.06.99)

(30) Priority Data:

09/098,892	17 June 1998 (17.06.98)	US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, M/S UPAL01-521, Palo Alto, CA 94303 (US).

(72) Inventor: UHLER, Stephen; Sun Microsystems, Inc., 901 San Antonio Road, M/S: UPAL01-521, Palo Alto, CA 94303 (US).

(74) Agents: HECKER, Gary,.A. et al.; Hecker & Harriman, Suite 2300, 1925 Century Park East, Los Angeles, CA 90067 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD AND APPARATUS FOR AUTHENTICATED SECURE ACCESS TO COMPUTER NETWORKS

(57) Abstract

Embodiments of the invention comprise a method and apparatus for authenticating secure access to computer networks. Embodiments of the invention control and manage access to a computer intranet from an extranet. Access to the intranet is allowed such that specified packets are permitted to penetrate the intranet's gateway and transmitted to a reverse proxy. The reverse proxy configurations authenticate a user, provide logging (e.g., intranet access), forward user credentials to intranet applications and provide a mapping between external references to intranet resources and their internal references. Mappings can be expressed literally or as a pattern expression.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD AND APPARATUS FOR AUTHENTICATED SECURE ACCESS TO COMPUTER NETWORKS

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

This invention relates to the field of computer software, and, more specifically, to secure and authenticated computer system access.

Portions of the disclosure of this patent document may contain material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office file or records, but otherwise reserves all copyright rights whatsoever. Sun, Sun Microsystems, the Sun logo, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

2. BACKGROUND ART

Computing devices can be connected via a communications network to transmit information between them and/or share peripheral devices (e.g., printers and storage devices) that are connected to the communications network. A communications protocol specifies a convention for communication over the network. For example, a communications protocol can identify the format for messages sent over the network.

The Internet is an example of a world wide communications network comprised of various physical networks that interconnect computing devices. The Internet is comprised of many physical networks that interconnect computing devices. For example, a personal computer in a user's home can be connected via one or more networks that comprise the Internet to a computer system regardless of either's location to gain access to information that is resident on that computer system. A user's request can be transported via the Internet's networks to the computer system. A response from the computer system can be transmitted to the user via the Internet.

The Transport Control Protocol/Internet Protocol (TCP/IP) are the basic communications protocol for transmitting information over Internet. A communications protocol typically defines the format for a packet, or bundle, of data that is to be transmitted. A packet usually includes control information (e.g., destination, origin, packet length, etc.), the data to be transmitted and error detection and correction. Other communications protocols, such as Hypertext Transmission Protocol (HTTP) and File Transfer Protocol (FTP), are built on top of TCP/IP. Resources (e.g., servers, services, program code, and files) are accessible via the Internet and are typically referenced by a universal resource locator (URL) that identifies the resource, the location of the resource and the protocol used to obtain the resource. A URL is mechanism by which a resource can be identified in a request. HTTP and FTP are mechanisms by which the request is communicated.

One example of a resource that can be requested by specifying a URL is a Hypertext Markup Language (HTML) document that defines a page of graphic content. HTML is a language that can be used to specify the page's graphic user interface (GUI) elements. An HTML document is transmitted via the HTTP communications protocol to a client that is running a software package referred to as a browser. A browser provides a GUI to display a page of information that is defined using HTML. The browser parses the HTML statements to generate and display the page's GUI elements in the browser's display area. The browser further provides a mechanism for the user to input information and/or to submit a request which the browser forwards, via the Internet, to the appropriate Internet server using a communications protocol such as HTTP.

A communications network can be characterized as either an external network (i.e., extranet) or an internal network (intranet). An extranet is a communications network that is considered to be external with reference to a given organization or entity. A network may be considered to be external simply because it is under another's administration and control. When viewed from a corporation's perspective, for example, the Internet is comprised of networks that are examples of extranets.

Similarly, a communications network to which access is controlled or restricted is an internal network (or intranet). An intranet operates over a physical network that is under a given entity's administrative control.

An intranet can be connected to an extranet via a physical connection such as a modem and telephone line. Routing hardware and/or software is used to route packets between the intranet and an extranet via a physical connection. A gateway which is comprised of hardware and/or software is typically used to act as an entrance and exit into a communications network. For example, an intranet can use a gateway through which packets directed to and from the intranet must pass. A gateway can further perform conversions between otherwise incompatible communications networks.

An entity may wish to limit the packets that are allowed access to its intranet. For example, an entity may wish to limit entry to information that is resident on its intranet such that it is not accessible to extranet users (e.g., an Internet user unaffiliated with the entity). However, current techniques for controlling external entry to an intranet are intended to prohibit external access to an intranet, or introduce a potential for breach of the intranet's security.

A firewall is one example of a technique that implements a restrictive and controlled access approach to an intranet (e.g., between an intranet and an extranet). A firewall is hardware and/or software (typically considered part of a gateway) that examines packet data to determine whether the packet should be forwarded to/from the intranet. The firewall identifies the destination and/or origin addresses to determine whether to forward the packet, for example. Where the firewall has been configured to stop the entry of packets from certain sources, the firewall examines the origin of the message and does not forward a message from an unauthorized source. If, for example, the firewall is configured to stop Internet packets from entering an intranet, the firewall blocks packets whose origin is the Internet. Similarly, a firewall can be used to intercept and stop a packet that is destined for an unauthorized destination (e.g., an extranet).

The restrictive and controlled access that is enforced by a firewall is advantageous because it reduces the chance of a security breach by an unauthorized user who otherwise might gain access to the intranet. However, a firewall prohibits an intranet user from accessing the extranet via the intranet.

To allow a user to gain access to the Internet from the intranet, a proxy server can be installed on the intranet which has access to both the intranet and the Internet. A proxy server acts as a proxy to forward requests on another's (e.g., an application's or user's) behalf. A proxy server forwards a message without modifying its content.

A proxy server typically performs application-level filtering of messages. That is, a proxy server examines application-level messages to determine whether and to whom the message should be forwarded. A proxy server can be used, for example, to forward information between two applications (or users) that reside on different intranets or between an intranet application (or user) and an extranet (e.g., the Internet). To access the Internet, for example, an intranet user sends a request directed to the Internet to the proxy server which forwards the request unchanged to the Internet.

Neither the firewall nor a proxy server allow access by an authorized user attempting to gain access to the corporation's intranet from outside the intranet. The purpose of the firewall is to prohibit external access. A proxy server's purpose is to facilitate access within the intranet. One way of allowing access by an authorized external user, is to eliminate the firewall. However, this would open the corporation's intranet to unauthorized users as well.

A virtual private network approach has been used in cooperation with the firewall to allow an external user to access the intranet. An IP packet is enclosed within another IP packet by the virtual private network software that is running on a computer system on the extranet. The outer packet is addressed to an intermediate destination within the intranet. The firewall is configured to allow IP packets that are destined for the intermediate destination in the intranet. When the packet is received by the intermediate destination, it extracts the inner IP packet that contains the true intranet destination. The IP packet is then forwarded by the intermediate destination.

Similarly, an IP packet that originates on the intranet and destined for the extranet is enclosed within an outer IP packet that identifies a permissible origin (i.e., an origin from which the firewall is configured to allow an IP packet to be transmitted to the extranet). The firewall examines the outer IP packet's origin address and determines that it is permissible to forward the IP packet to the extranet from that origin.

The virtual private network includes software that is running on the extranet client and the intranet (e.g., the destination server). The virtual private network running on the extranet client encloses the original IP packet in another IP packet that is addressed to a permissible destination within the intranet. The inner IP packet can identify any destination address on the intranet. Thus, an unauthorized user that gains access to a virtual private network client has uninhibited access to the intranet. Thus, a disadvantage of this approach is that the intranet is only as secure as the user's workstation. Therefore, a virtual private network is optimally used with a secure workstation that communicates with the intranet via a leased, or dedicated, telephone line. The virtual private network is clearly not optimal where the workstation is a laptop that could be left at an unsecured location or other computer system that is susceptible to public access, for example.

Another approach to accessing the intranet is referred to as web tunneling. or a web tunnel. The web tunnel restricts the manner in which an Internet user accesses the Internet and is, therefore, limited in its applicability. In this approach, a user must configure the client browser to direct all of its intranet requests directly to the web tunnel. The web tunnel approach cannot be used where the user is attempting to access one intranet from inside another intranet, or where the user must access the Internet via a proxy server. Figure 8 provides a block diagram of the web tunnel approach.

Figure 8 depicts web tunnel 800 that includes authenticator 804A, redirector 804B and proxy 804C. The user must configure client 802 to send its requests for intranet 820 directly to redirector 804B. Redirector 804B redirects a request to either authenticator 804A or proxy 804C components of web tunnel 800. Authenticator 804A produces material that is used to authenticate client 802 to proxy 804C. Proxy 804C performs the function of receiving requests for web servers 806 and 808 and forwarding requests to them.

When redirector 804B receives a URL from client 802, redirector 804B packages the URL inside another URL that identifies either authenticator 804A or 804C. For example, if the user sends a URL for a document that resides on a server inside intranet 820, redirector 804B appends the user's URL to proxy 804C's URL.

The transaction steps are illustrated in Figure 8. At step 1, client 802 sends an HTTP URL, "http://hr.acme.com" to redirector 804B. Redirector 804B generates a redirected URL, "https://tunnel.acme.com/hr.acme.com" to specify proxy 804C of web tunnel 800. At step 2, redirector 804B sends the redirected URL back to client 802. Client 802 must then forward the new, redirected URL to web tunnel 800 (i.e., proxy 804C). Proxy 804C receives and processes the redirected URL.

In the web tunnel approach, the user must repeat a request where the request identifies an intranet resource by its intranet URL. That is, before an internal URL can be forwarded to the intranet via web tunnel 800, it must first be packaged (by redirector 804B) inside another request that is directed to proxy 804C and then resent by the user. Further, the web tunnel approach requires that the user be aware of the URLs used to identify resources inside the intranet. This is disadvantageous since the user may not be aware of the actual URL of intranet resources. This also exposes the intranet structure to an external user.

The web tunnel approach requires a configuration that restricts the type of user that can use the approach. That is, the web tunnel approach requires that there be no proxies between client 802 and web tunnel 800. This is unrealistic since, as described above, most access schemes use a proxy server as a conduit for transmissions between the client and the Internet.

The web tunnel approach cannot be used by a corporate intranet user who is required to use a proxy server to access the Internet, for example. The web tunnel approach requires that the client browser be configured directly to web tunnel 800. The structure of the intranet is such that a client request could be forwarded to an ultimate destination via multiple computer networks. Each of these computer networks may require that a proxy server be used to direct the request to the next Internet destination. The web tunnel approach is incapable of functioning in this type of environment. It is limited to direct connection between the client and the web tunnel that services the request's final destination. That is, the client must be directly connected to the final destination's web tunnel mechanism. There is no ability for one or more proxies to be positioned between the client and the web tunnel.

SUMMARY OF THE INVENTION

Embodiments of the invention comprise a method and apparatus for secure authenticated access to computer networks. Embodiments of the invention control and manage access to an intranet from an extranet. Access to the intranet is allowed such that specified packets are permitted to penetrate the intranet's gateway. Embodiments of the invention can be used to limit access to the intranet to specific types of messages (e.g., messages received from browser software running on a client workstation) transmitted via the Internet, for example.

Embodiments of the invention offer multi-tiered access control. A user is authenticated before being allowed access to an intranet. Further, access privileges associated with an authenticated user are identified and used to determine the resources that an authenticated user is permitted to access. A user's authentication information is retained and provided to intranet applications based on an application's requirements. Thus, the user's credentials can be used to sign the user onto multiple intranet applications.

A logging facility is offered in embodiments of the invention for logging information associated with intranet accesses. Embodiments of the invention can log internal errors, configuration errors, login attempts, login failures, session time-outs, session terminations, and performance metrics.

Embodiments of the invention provide a transparent application pass-through. An external reference is used for an intranet resource. The external reference need not be the same as the actual reference of the intranet resource (as used in the intranet).

A mapping is used to associate the external, or virtual, reference with the actual, intranet reference. Mapping between external and internal references is performed transparently to the user and/or applications executing on the intranet. Embodiments of the invention map intranet references to external references before the reference is sent to the user. Conversely, an external reference that is received from the user can be translated to the intranet reference. There is no need for an external user to be aware of a resource's actual, intranet reference. This is advantageous for both facilitating an authenticated user's access and shielding the intranet's structure.

Embodiments of the invention include configuration information that identifies the hosts (e.g., servers) that are allowed to be mapped between internal and external references. An internal reference to an intranet resource is re-written as an external reference when a reference to the resource is sent out to an external computer.

Thus, embodiments of the invention map external resource references to internal resource references. Each mapping consists of an internal mapping entry and an associated external mapping entry. A mapping entry can be specified using literal expressions or pattern expressions. When an internal reference is being re-written to an external reference, internal mapping entries (e.g., literal and/or pattern expressions) associated with a user are compared to the internal reference to identify a match. If a match exists, the internal reference is re-written as an external reference using the external mapping entry associated with the matched internal mapping entry. A similar approach is taken to map an external reference to an internal reference.

The mappings that are associated with a user are identified in embodiments of the invention based on a user's credentials. A user's credentials identify the access privileges for the user. If, for example, a user's credentials identify the user as an employee, the user can be given those access privileges that are given to an employee. A set of mappings can be associated with the employee privilege. If the user has multiple privileges each of which has a set of mappings, the multiple sets of mappings can be combined to generate a complete set of mappings for the user. Later processed mappings can supplement or modify previously-processed mappings, for example.

When the user references an intranet resource, the mappings associated with the user can be searched to find the reference submitted by the user. If the reference is not found in the user's mappings, access is denied. If a response (or other message) is returned to the user, internal references are re-written to external references according to the user's set of mappings. It is possible, however, that the set of mappings may specify that the internal reference is to be forwarded without any modification.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of one embodiment of a computer system capable of providing a suitable environment for an embodiment of the invention.

Figure 2A provides a model used in one or more embodiments of the invention.

Figure 2B illustrates a model implementation of a reverse proxy as a component of an intranet gateway according to one embodiment of the invention.

Figure 2C illustrates configuration alternatives according to one or more embodiments of the invention.

Figure 3 illustrates a request processing model according to an embodiment of the invention.

Figure 4 illustrates a login and authentication model according to an embodiment of the invention.

Figure 5 provides an request processing process flow according to an embodiment of the invention.

Figures 6A-6B provide an authentication process flow according to an embodiment of the invention.

Figure 7 provides a request processing process flow according to an embodiment of the invention.

Figure 8 provides a block diagram of the web tunnel approach.

DETAILED DESCRIPTION OF THE INVENTION

A method and apparatus for authenticated secure access to computer networks is described. In the following description, numerous specific details are set forth in order to provide a more thorough description of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known features have not been described in detail so as not to obscure the invention.

Embodiments of the invention comprise a method and apparatus for authenticated secure access to computer networks. Embodiments of the invention control and manage access to a computer intranet from an extranet. Access to the intranet is allowed (through any number of firewalls or proxies) such that specified packets are permitted to penetrate the intranet's gateway. Embodiments of the invention can be used to limit access to the intranet to specific types of packets (e.g., packets received from browser software running on a client workstation) transmitted via an extranet.

Embodiments of the invention offer multi-tiered access control. A user is authenticated before being allowed access to an intranet. Further, access privileges associated with an authenticated user are identified and used to determine the resources that the user is permitted to access. A user's authentication information is retained and provided to applications that exist on the intranet based on an application's requirements. Thus, the user's single sign-on information can be used to sign the user onto multiple intranet applications.

A logging facility is offered in embodiments of the invention for logging information associated with intranet accesses. Embodiments of the invention provide a transparent application pass-through. A reference is used for external accesses of an intranet resource. The external reference need not be the same as the actual reference of the intranet resource (as used in the intranet). A mapping is used to associate the external, or virtual, reference with the actual, intranet reference. There is no need for an external user to be aware of a resource's actual, intranet reference. This is advantageous for both facilitating an external user's access and shielding the intranet's structure.

Embodiments of the invention use the mapping of external references to internal references to transform an intranet reference to an external reference before the reference is sent to the user. Conversely, an external reference that is received from the user can be translated to the intranet reference. Mapping between external and internal references can be performed on those resources that exist on the intranet to which the user has authorization to access, for example. Mapping between external and internal references is performed transparently to the user and/or applications executing on the intranet.

Embodiments of the invention include mapping information that identifies the internal and external references used for intranet resources (such as hosts or servers) for each user accessing the intranet from an extranet. An internal reference to an intranet resource's is re-written as an external reference as the reference is sent out to an extranet. When an external reference to an intranet resource is received, the external reference can be translated to the intranet resource's internal reference. The mapping information can identify resources using literal or pattern expressions. A pattern expression can be used to identify a group of resources, for example.

A description is given below of an embodiment of a computer apparatus suitable for providing an execution environment for the software apparatus of the invention.

Embodiment of Computer Execution Environment (Hardware)

An embodiment of the invention can be implemented as computer software in the form of computer readable program code executed on a general purpose computer such as computer 100 illustrated in Figure 1. A keyboard 110 and mouse 111 are coupled to a bi-directional system bus 118. The keyboard and mouse are for introducing user input to the computer system and communicating that user input to processor 113. Other suitable input devices may be used in addition to, or in place of, the mouse 111 and keyboard 110. I/O (input/output) unit 119 coupled to bi-directional system bus 118 represents such I/O elements as a printer, A/V (audio/video) I/O, etc.

Computer 100 includes a video memory 114, main memory 115 and mass storage 112, all coupled to bi-directional system bus 118 along with keyboard 110, mouse 111 and processor 113. The mass storage 112 may include both fixed and removable media, such as magnetic, optical or magnetic optical storage systems or any other available mass storage technology. Bus 118 may contain, for example, thirty-two address lines for addressing video memory 114 or main memory 115. The system bus 118 also includes, for example, a 32-bit data bus for transferring data between and among the components, such as processor 113, main memory 115, video memory 114 and mass storage 112. Alternatively, multiplex data/address lines may be used instead of separate data and address lines.

In one embodiment of the invention, the processor 113 is a microprocessor manufactured by Motorola, such as the 680X0 processor or a microprocessor manufactured by Intel, such as the 80X86, or Pentium processor, or a SPARC microprocessor from Sun Microsystems, Inc. However, any other suitable microprocessor or microcomputer may be utilized. Main memory 115 may be comprised of dynamic random access memory (DRAM). Video memory 114 is a dual-ported video random access memory. One port of the video memory 114 is coupled to video amplifier 116. The video amplifier 116 is used to drive the cathode ray tube (CRT) raster monitor 117. Video amplifier 116 is well known in the art and may be implemented by any suitable apparatus. This circuitry converts pixel data stored in video memory 114 to a raster signal suitable for use by monitor 117. Monitor 117 is a type of monitor suitable for displaying graphic images.

Computer 100 may also include a communication interface 120 coupled to bus 118. Communication interface 120 provides a two-way data communication coupling via a network link 121 to a local network 122. For example, if communication interface 120 is an integrated services digital network (ISDN) card or a modem, communication interface 120 provides a data communication connection to the corresponding type of telephone line, which comprises part of network link 121. If communication interface 120 is a local area network (LAN) card, communication interface 120 provides a data communication connection via network link 121 to a compatible LAN. Communication interface 120 could also be a cable modem or a wireless interface. In any such implementation, communication interface 120 sends and receives electrical, electromagnetic or optical signals which carry digital data streams representing various types of information.

Network link 121 typically provides data communication through one or more networks to other data devices. For example, network link 121 may provide a connection through local network 122 to host computer 123 or to data equipment operated by an Internet Service Provider (ISP) 124. ISP 124 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 125. Local network 122 and Internet 125 both use electrical, electromagnetic or optical signals which carry digital data streams. The signals through the various networks and the signals on network link 121 and through communication interface 120, which carry the digital data to and from computer 100, are exemplary forms of carrier waves transporting the information.

Computer 100 can send messages and receive data, including program code, through the network(s), network link 121, and communication interface 120. In the Internet example, server 126 might transmit a requested code for an application program through Internet 125, ISP 124, local network 122 and communication interface 120. In accord with the invention, one such downloaded application is the method and apparatus for authenticated secure access to computer networks described herein.

The received code may be executed by processor 113 as it is received, and/or stored in mass storage 112, or other non-volatile storage for later execution. In this manner, computer 100 may obtain application code by way of a carrier wave.

Application code may be embodied in any form of computer program product. A computer program product comprises a medium configured to store or transport computer readable code, or in which computer readable code may be embedded. Some examples of computer program products are CD-ROM disks, ROM cards, floppy disks, magnetic tapes, computer hard drives, servers on a network, and carrier waves.

The computer system described above is for purposes of example only. An embodiment of the invention may be implemented in any type of computer system or programming or processing environment including, but not limited to, embedded systems and/or "thin" access devices such as a web phone.

Model

Embodiments of the invention comprise software configured to allow authenticated, secure access to a computer network such as an intranet. Figure 2A provides a model used in one or more embodiments of the invention.

Client 202 comprises a computer system connected to an extranet (e.g., extranet 244). Client 202 can be, for example, running browser software and capable of connecting to the Internet in some manner. For example, client 202 can be an individual user that directly connects to the Internet via an Internet Service Provider. Alternatively, client 202 can be a corporate user that accesses the Internet via a corporate intranet and a proxy server running on the intranet, for example.

A request generated by client 202 is transmitted via line 220 (i.e., the user's connection to the Internet). Where the request is encrypted, line 220 can represent a secure socket connection such as that provided by Netscape's Secure Socket Layer (SSL) mechanism. Line 220 can represent an immediate connection between client 202 and reverse proxy 204. Alternatively, line 220 can represent an indirect connection between client 202 and reverse proxy 204. That is, a communication between client 220 and reverse proxy 204 can be made via intermediate networks and any number of proxy (or other) servers at each intermediate point.

In one or more embodiments of the invention, reverse proxy 204 comprises program code configured to control access to an intranet from an external source (an extranet such as the Internet). In embodiments of the invention, reverse proxy 204 is configured to authenticate user(s), facilitate single sign-on for users, perform logging functions, control access to resources (e.g., intranet resources), and/or facilitate references to authorized resources (e.g., intranet resources).

Application servers 212A-212C are examples of resources of an intranet. A request that is initiated by client 202 may be processed by one of application servers 212A-212C, for example. A request may be for a Web page that resides on application server 212A, for example. To further illustrate, the request can be for program code (e.g., an applet) to be downloaded to client 202 or for invocation of program code on one of application servers 212A-212C, for example.

Reverse proxy 204 comprises program code that is configured to, among others, authenticate the user of client 202. Reverse proxy 204 interacts with authentication server 208 to authenticate the user. Information entered by the user during the authentication is retained and can be forwarded by reverse proxy 204 to authenticate the user to an application running on one of application servers 212A-212C. Reverse proxy 204 can retain user information using state server 206, for example. Lines 232 and 222 represent communications between reverse proxy 204 and authentication server 208 and reverse proxy 204 and state server 206 (respectively) which can be achieved using a remote procedure call (RPC) mechanism, for example.

Reverse proxy 204 further ensures that an authenticated user is only permitted to access authorized resources on the intranet. Authentication server 208 forwards an authenticated user's access privileges to reverse proxy 204. Reverse proxy 204 determines what a user is permitted to access using the access privileges received from authentication server 208.

Access privileges can be comprised of a list of intranet resources to which a user is allowed access. In addition (or as an alternative), access can be expressed using a generic identification or pattern that is compared to a resource's reference to determine whether the reference and pattern match. For example, a group of resources that start with the letter "a" and include the letter "g" might be expressed as "a*g*", for example, where "*" is a wild card for none or more characters. All resource references that begin with an "a" and include a "g" match this pattern. Reverse proxy 204 can perform a matching operation with the pattern and the resource identifier supplied by the user to determine whether the user is permitted to access the resource. In one or more embodiments of the invention, the generic identification or pattern can be used to determine whether an external reference is translated to its internal reference and vice versa.

If reverse proxy 204 determines that a user's request is directed to a permitted resource, reverse proxy 204 forwards the request via the intranet to a destination to access the resource. For example, a request from a user who is authorized to access application server 212A is forwarded by reverse proxy 204 to proxy server 210 (via line 234) as plain text. Proxy server 210 directs the request to application server 212A, via line 224. Application servers 212A-212C receive a request from proxy server 210 via lines 224, 226 and 228 which represent a plain text transaction, for example.

Reverse proxy 204 retains a user's credentials (e.g., via state server 206) that can be forwarded to an application that executes on the intranet. The application need not request initial sign-on information from the user. Further, the user does not have to perform multiple sign-ons for each intranet application. Instead, the information received from the user by reverse proxy 204 can be forwarded to an intranet application. Thus, there is a single sign-on and authentication for a user.

Reverse proxy 204 can facilitate logging. For example, reverse proxy 204 can log internal errors, configuration errors, login attempts, login failures, session time-outs, session terminations, and performance metrics.

In the Internet, or World Wide Web (WWW), a user typically identifies a resource (e.g., a file, application or application server) using a universal resource locator (URL) specification. A URL identifies the name of the resource, its location and the protocol used to obtain it. For example, a URL identifies the server that can produce the resource (e.g., a file) and the protocol used to access the server. For example:

http://www.sun.com/index.html

identifies the http protocol and a file (i.e., an Hypertext Markup Language, HTML, file named index.html) on a server identified as "www.sun.com."

Reverse proxy 204 maps an external reference for an intranet resource to its internal reference (i.e., the name used on the intranet to identify the resource). Thus, for example, an extranet user can identify an internal resource even if the user does not know the intranet resource's internal reference. Further, it may be desirable to mask the actual intranet resource reference from the extranet user.

A different mapping between external and internal references provides the ability to associate an external reference with different internal references.

When one mapping is used, the external reference can map to a first internal resource. The same external reference can map to a second internal resource when a different mapping is used.

The mapping mechanism of reverse proxy 204 can be used to specify a user's the access privileges. For example, the absence of a mapping to an internal resource in a set of mappings for a user signifies that the user is not authorized to access the internal resource.

A web server can be used to forward requests for web documents to a content server and to forward responses received from a content server to a client browser, for example. Reverse proxy 204 can be implemented as a common gateway interface (CGI) script that runs on an intranet's web server. Alternatively, where the web server provides a mechanism for creating plug-ins to the web server's program code, reverse proxy 204 can be implemented as a plug-in of the web server. If reverse proxy 204 is implemented as a CGI script or as a plug-in, the web server forwards all requests directed to the intranet to reverse proxy 204 for processing. Reverse proxy 204 forwards a response to the web server for forwarding to the extranet (e.g., a client browser). As yet another alternative, reverse proxy 204 can be implemented as a stand alone component of the architecture. In this case, reverse proxy 204 can also act as a web server.

According to an embodiment of the invention, an implementation (e.g., CGI script, plug-in, or stand alone implementation) of reverse proxy 204 is included as a part of the gateway between the intranet and external networks such as the Internet. Figure 2B illustrates a model implementation of a reverse proxy as a component of an intranet gateway according to one embodiment of the invention. Intranet 248 consists of proxy server 210, authentication 208 and application servers 212A-212C, as described in conjunction with Figure 2A.

Intranet 248 is a computing device, and/or network, access to which is to be restricted or controlled. Client 202 is resident on extranet 244. Extranet 244 is a computing device and/or network whose access to intranet 248 is to be controlled or restricted. Extranet 244 can be, for example, the various communications networks that comprise the Internet. Reverse proxy 204 is configured to control access to intranet 248 from extranet 244.

Gateway (i.e., intranet gateway) 246 comprises firewalls 240 and 242 and reverse proxy 204. Firewalls 240 and 242 examine received packets to determine whether the packet should be allowed access to intranet 248. A firewall is a software application that typically examines a packet's header to determine the packet's type, the sender and the intended recipient. The access criteria made known to the firewall (e.g., via configuration information) is used by the firewall to determine whether the information contained in the packet satisfies the criteria.

In this embodiment of the invention, firewall 240 is configured to allow packets (e.g., IP packets that are addressed to reverse proxy 204). The packet is forwarded to reverse proxy 204 which ensures that the packet is compliant with the HTTP protocol. Firewall 242 can be configured to allow messages (e.g., RPC messages) originating from reverse proxy 204 and bound for either proxy server 210 or authentication server 208.

It should be apparent that the use and number of firewalls and proxies (e.g., HTTP proxies) used in conjunction with embodiments of the invention can vary. Firewalls merely provide additional checkpoints for filtering packets directed to the intranet. Reverse proxy 204 provides a checkpoint that screens received packets and can be configured to accept certain packets and deny access to the rest. Thus, intranet gateway 246 can be comprised of reverse proxy 204, or reverse proxy 204 and one or more firewalls.

The model can be scaled and adapted to satisfy multiple configurations. Figure 2C illustrates configuration alternatives according to one or more embodiments of the invention. An extranet (e.g., extranet 244 of Figure 2B) includes client 202. Box 276 represents the intermediate connections (e.g., proxy servers such as proxy servers 230 and/or 240, or other intermediate servers, computing devices and/or networks) between client 202 and an instance of reverse proxy 204. Box 276 can contain none or more intermediate connections. Thus, client 202 can be directly connected to an instance of reverse proxy 204 (e.g., reverse proxy 224), or connected to an instance of reverse proxy 204 via one or more intermediate servers. A request from client 202 can be routed via intermediate servers to reverse proxy 224.

Multiple instances of reverse proxy 204, reverse proxies 224 and 234, can control access to an intranet. Reverse proxies 224 and 234 can authenticate the user of client 202 via an authentication server (e.g., authentication servers 228 and 218, respectively). In Figure 2C, reverse proxies 224 and 234 use different authentication servers (authentication servers 228 and 218, respectively). However, they can use the same authentication server.

Box 286 indicates that there can be one or more instances of reverse proxy 204 to control access to an intranet. Each instance of reverse proxy 204 can be configured to authenticate a user, support single sign-on, perform logging and access control and map external references to internal references of intranet resources, for example.

A first authentication can be performed by reverse proxy 224. If the authentication is successful, the request is processed by reverse proxy 224. Reverse proxy 224 can forward the request to application 212B via proxy 250 where the request is directed to application 212B, for example. Alternatively, reverse proxy 224 can forward the request to reverse proxy 234 if, for example, the request is directed to application server 212A. Reverse proxy 234 can perform a second authentication, if desired.

An instance of reverse proxy 204 can be directly connected to an intranet resource as illustrated by reverse proxy 234 (e.g., reverse proxy 234 is directed connected to application server 212A). Alternatively, an instance of reverse proxy can be indirectly connected to an intranet resource via, for example, one or more proxy servers. For example, reverse proxy 224 is connected to application server 212B via proxy server 250.

Request Processing

Reverse proxy 204 interacts with client 202 to receive and process a user request. Reverse proxy 204 enforces access privileges associated with a user. Further, reverse proxy 204 can perform a logging function. Figure 3 illustrates a request processing model according to an embodiment of the invention. (Embodiments of the invention comprising user authentication are described below.) The request processing described in Figure 3 assumes that the user of client 202 has been authenticated and reverse proxy 204 is aware of the access privileges associated with the user of client 202.

Client 202 transmits a request to reverse proxy 204. An authenticated user's authorized access request is forwarded to the intranet resource. For example, a request from an authenticated user to access application server 212A (e.g., an application that is running on application server 212A) is forwarded to application server 212A via proxy server 210 by reverse proxy 204.

Reverse proxy 204 ensures that the user has sufficient access privileges to access the intranet resource referenced in the user's request. That is, reverse proxy 204 examines a list of authorized resources associated with a user to determine whether the user has been authorized to access the referenced resource. Therefore, if the request originates from an authenticated user who has sufficient access privileges to access the reference resource, reverse proxy 204 changes the resource reference to its internal reference and forwards the request to the intranet resource using its internal reference.

Mapping

Reverse proxy 204 maps external resource references contained in the request to actual intranet resource references. Conversely, reverse proxy 204 maps internal resource references contained in a message transmitted from the intranet (e.g., a response to a request) to their corresponding external resource references. The mappings that are performed by reverse proxy 204 are performed on each transmission as it is transmitted between the intranet and an extranet.

For example, a transmission such as a URL request can be sent to the intranet from an extranet. The transmission can be, for example, a user's request in the form of a URL that comprises an external reference to an

HTML page that resides on a server on the intranet. Reverse proxy 204 translates the external reference to an internal reference, if a mapping exists for the user between the external reference and its internal reference.

Conversely, a reference to an internal resource may be bound for an extranet. For example, a response to a request generated on the intranet, such as a response generated by an application running on application server 212A, can contain an internal reference to an intranet resource. The request is forwarded by application server 212A to proxy server 210 which forwards it to reverse proxy 204. Reverse proxy 204 translates internal references contained to the response to external references, using a set of mappings associated with the user, before forwarding the response to client 202. Thus, for example, if the response is an HTML page that contains hyperlinks to other HTML pages that reside on the intranet, reverse proxy 204 translates the hyperlink references to external references using the user's set of mappings.

In one or more embodiments of the invention, the external reference identifies reverse proxy 204. Thus, when a user requests a intranet resource by its external reference (e.g., selects a hyperlink that has been re-written by reverse proxy 204), the request is forwarded to reverse proxy 204. Reverse proxy 204 can attempt to re-write the external reference using the set of mappings associated with the requesting user.

In an embodiment of the invention, reverse proxy 204 refers to a table that identifies a mapping between an intranet resource's external reference and its internal reference. The following provides an example of a table of mappings according to an embodiment of the invention:

External Reference	Internal Reference
help	{internal.helpServer:8015 /help/ }
faq	{internal.faqServer:8015 /notes/ }
nametool	{internal.nameServer:8015 /name.subst/ }
calendar	{internal.calServer:8015 /calendar/ }
java	internal.java:80
([a-zA-Z]+)\.([a-zA-Z]+)(:80)?	\1.\2:8080

The internal reference identifies an intranet resource (e.g., a server named "internal.helpServer"). The intranet resource's internal reference is mapped to an external reference for the intranet resource. In an embodiment of the invention, an intranet resource's internal reference is re-written (as an external reference) to appear as though it is on reverse proxy 204. The external reference includes a reference to reverse proxy 204.

Reverse proxy 204 substitutes the external reference in place of the intranet resource's intranet reference in a transmission (e.g., a response) that is sent to client 202. For example, a link in an HTML document that is expressed as an intranet reference to an index.html file in the help directory of the internal.helpServer server can be translated to a "help" external reference by reverse proxy 204. An intranet reference of "internal.helpServer:8015/help/index.html" can be translated to a "help/index.html" reference, for example, to present to the user.

Conversely, when a request is received from the user, an external reference can be mapped to its intranet reference. If, for example, the user selects the "help/index.html" link in the HTML document, reverse proxy 204 translates the "help/index.html" to its intranet reference before it forwards the request. Thus, the "help/index.html" is translated to "internal.helpServer:8015/help/index.html" by reverse proxy 204, for example. Reverse proxy 204 forwards the request to the "internal.helpServer" server via port 8015. Thus, reverse proxy 204 forwards a request including its intranet references to the intranet (e.g., initially to proxy server 210).

The first five entries in the mappings table contain literal expressions for both the internal and external references. A mapping can also be expressed in terms of pattern expressions. If a pattern expression is given, it is used to translate references that match the pattern expression. For example, if a pattern expression is used in the external reference mapping entry (e.g., the left-hand column in the above mapping table), an external reference is compared to the external reference pattern expression. If the external reference matches the pattern, then the external reference can be translated to the internal reference mapping entry (e.g., the right-hand column in the above mapping table). Where the internal reference mapping entry is expressed in terms of a pattern expression, the pattern expression can be used to generate the internal reference.

For example, in the above mappings table, the last table entry includes a pattern expression for both the external and internal references. While any format can be used to express a pattern, the mappings table example uses regular expressions to describe a pattern. The external reference pattern expression example matches an external reference that contains a first set of alphabetic characters followed by a period (".") followed by a second set of alphabetic characters, and optionally followed by a ":80". Thus, external references such as "xyz.cde" or "xyz.cde:80" matches the external reference pattern expression. An external reference such as "xyz.cde.abc" does not match the external reference pattern expression.

If the external reference matches the external reference pattern expression, the external reference is translated into the internal reference. In the example given above, the internal reference is also expressed as a pattern. In this case, the internal reference pattern expression is used to translate the external reference into the internal reference. For example, the first set of alphabetic characters from the external reference becomes the first part of the internal reference followed by a "." followed by the second set of alphabetic characters from the external reference. If the external reference contained an ":80" character string, it is replaced by an ":8080" character string. If, for example, the external reference was "xyz.cde:80", it is translated to an "xyz.cde:8080" internal reference.

If an external reference does not match the external reference pattern expression, it is not translated into an internal reference. If the external reference cannot be translated into an internal reference using another external reference entry in the mappings table, the external reference cannot be used to access an internal resource. Thus, access to the internal resource should be denied. Thus, for example, the external reference of "xyz.cde.abc" does not match any of the external reference entries in the mappings table, is not translatable to an internal resource using this mappings table and access to an internal resource is therefore not possible. The user could be informed, for example, that there is no such resource with the name specified by the external reference.

Thus, reverse proxy 204's mapping mechanism can be used to enforce an access policy for external users. Further, using the mapping mechanism, a user of client 202 need not have knowledge of the actual internal reference for an intranet resource. This simplifies a user's access. Further, this level of redirection allows resources to be moved to different sites on the intranet without requiring the user to be aware of the move. In addition to simplifying a user's access, there are security benefits to shielding the structure of the intranet from the user. Using external references that map to their corresponding intranet references has the effect of hiding the intranet's actual structure. This can limit a user's ability to successfully attempt an unauthorized access of the intranet, or a component of the intranet.

The same external reference can map to a different internal resource depending on which set of mappings are used. The set of mappings that are used for translating references for a given user can be specified in the user's credentials. The same external reference used by two different users can translate into different internal references where the two users are assigned a different set of mappings.

User Credentials

Authentication server 208 returns a user's credentials for an authenticated user. In an embodiment of the invention, a user's credentials identify the set of mappings for the user. The user's credentials can specify a set of privileges (e.g., a user may have the privileges associated with an employee, a member of a given department, and/or a member of management of a given department). A user's privileges are used to identify the set of mappings for the user. Multiple sets of mappings can be combined to construct a master set of mappings for the user. For example, where each privilege included in a user's credentials identifies a different set of mappings, the sets of mappings can be combined to create one set of mappings for a user. The sets of mappings can be combined such that a mapping can be added based on one set of mappings and deleted or modified based on another (e.g., subsequently processed) set of mappings.

A user's credentials can be used to authenticate the user to a given intranet resource (e.g., an application). In an embodiment of the invention, reverse proxy 204 forwards the user's credentials (e.g., using one or more credential forwarding mechanisms available in the HTTP protocol). Reverse proxy 204 can send the user's credentials in addition to forwarding the user's information (e.g., userid) to the intranet. For example, an application that executes on application server 212A may require a userid to authenticate a user before the user's request can be processed by the application. Reverse proxy 204 forwards the request and credentials (that includes a userid) to application server 212A via proxy server 210.

User Login and Authentication

Reverse proxy 204 interacts with authentication server 208 to authenticate a user and retrieve a user's access privileges. A user's access privileges are used by reverse proxy 204 to determine whether a request to access an intranet resource is authorized. An authenticated user's authorized access request is forwarded to the intranet resource. Figure 4 illustrates a login and authentication model according to an embodiment of the invention.

To access the intranet initially, the user sends a request to reverse proxy 204. Since it is the user's initial request to the intranet, reverse proxy 204 requests user login information and authenticates the user. The authentication process, therefore, occurs upon a user's initial access to the intranet in one or more embodiments of the invention. User authentication can also take place at given times during a user's session.

Client 202 sends a request to reverse proxy 204. If client 202 has already logged on and has been authenticated, client 202 has a cookie (or piece of information) given to the user by reverse proxy 204 that client 202 can send to reverse proxy 204. If client 202 sends a valid cookie with the request, reverse proxy 204 can process the request as described in conjunction with Figure 3, for example.

If a valid cookie is not sent by client 202 along with the request, reverse proxy 204 assumes that the user is an authenticated user and initiates an authentication for the user. Reverse proxy 204 sends a request for user information to client 202. The request from reverse proxy 204 for user information (e.g., a userid) can be in the form of an HTML page that contains a set of prompts and input fields, for example. The user can enter the user information and submit the HTML page. Client 202 forwards the user information to reverse proxy 204. The requested user information is extracted from the HTML page.

Reverse proxy 204 forwards the userid to authentication server 208 and generates an information item or value (referred to as a cookie) such as a random number for the user. A user's cookie comprises a unique value for the user. A cookie can remain valid until the user logs off, or can become invalid prior to the user logging off, for example. A user's cookie can expire, for example, after the user has been logged on for a certain period of time, or if no transmissions are received from the user over a certain period of time.

Authentication server 208 generates a challenge (e.g., a randomly generated value) and forwards the challenge to reverse proxy 204. Reverse proxy 204 stores the challenge, cookie and userid. For example, reverse proxy 204 sends the challenge, cookie and userid to state server 206 for retention.

Reverse proxy 204 forwards the challenge to client 202 along with the cookie and instructions to forward the cookie with each subsequent transmission to reverse proxy 205. The user responds with a result that is generated from the challenge. For example, the user enters the challenge into computing device 404 (e.g., via a key pad of the computing device).

Computing device 404 is used to generate a unique value using input a portion of which is sent to client 202 by reverse proxy 204. An enigma card or smart card are examples of computing devices that execute an algorithm to compute a result based on input. These computing devices are available from various manufacturers.

According to this embodiment of the invention, the user inputs a challenge and another identifier (e.g., a personal identification number or PIN) into computing device 404. Computing device 404 generates a result which is referred to herein as the user's password, or result. In embodiments of the invention, at least one input value (i.e., the challenge) changes each time authentication is performed. Therefore, the result computed from the input will necessarily be different for each authentication. This creates a more secure environment, since the user's password (i.e., the result generated by the computing device) changes for each login.

In an alternate embodiment, X.509 digital certificates are used to authenticate the user. Some browsers have built-in capability to provide certificates such as those certificates that are based on the X.509 International Standards Organization (ISO) standard. When reverse proxy 204 requests a certificate, the web browser requests the user for a value referred to as a certificate identifier or pass phrase. The browser sends an X.509 certificate. Reverse proxy 204 can forward the certificate to authentication 208 to authenticate the certificate ID or pass phrase submitted by the user.

In an alternate embodiment, computing device 404 is not used to generate a result. Instead, the user simply enters a result (e.g., a password or PIN) that has been issued to the user. In this case, a challenge is not sent to the client 202. The user is authenticated on the basis of the userid and password values.

The user enters the result displayed by computing device 404 in a field of the browser software running on client 202 and client 202 forwards the result and cookie to reverse proxy 204. Reverse proxy 204 sends the cookie to state server 206 to retrieve the user's userid and challenge from storage. State server 206 retrieves the userid and challenge associated with the user's cookie and forwards them to reverse proxy 204.

Reverse proxy 204 sends the userid, challenge and result to authentication server 208. Authentication server 208 generates a result based on the challenge and verifies that the result received from the user is the expected result given the challenge provided to the user. If not, authentication server 208 forwards a rejection to reverse proxy 204. Reverse proxy 204 can forward the rejection to client 202. If the result is the expected result, authentication server 208 forwards the credentials associated with the userid to reverse proxy 204. The credentials can be, for example, a list of intranet resources that the user is given authority to access. The authorized resources can be itemized individually, or a pattern can be used to identify a set of resources whose identifiers (e.g., URLs) match the pattern.

Reverse proxy 204 uses the user's credentials to determine whether the user is authorized to access the intranet resource(s) identified in the user's request. If the user does not have the authority to access the intranet resource(s), a rejection message can be sent to client 202. If the user's credentials indicate that the user has authority to access the intranet resource(s), reverse proxy 204 forwards the request to intranet 248 (e.g., proxy server 210).

Request Processing Flow

In an embodiment of the invention, reverse proxy 204 processes requests received from both authenticated and unauthenticated users. Figure 5 provides a request processing process flow according to an embodiment of the invention. At step 502, client 202 sends a request to reverse proxy 204. At step 504, reverse proxy 204 makes a determination whether the user is an authenticated or unauthenticated user. If, for example, the request includes a valid cookie generated by reverse proxy 204, processing continues at step 518 to process the request. (Figure 7 provides an example of a process flow for processing an authenticated user's request.) Processing continues at step 502 to await another request.

If the request does not include a cookie or the cookie is not valid, processing continues at step 508 to authenticate the user. (Figures 6A-6B illustrate an authentication process flow according to an embodiment of the invention.)

At step 510, reverse proxy 204 makes a determination whether the user is an authorized user. For example, reverse proxy 204 examines the transmission from authentication server 208 to determine whether the transmission contains a rejection of the user or contains the user's credentials. If the transmission contains a rejection, processing continues at step 512 to send a rejection message to client 202 and processing continues at step 502 to await another request.

If authentication server 210 forwards the user's credentials, processing continues at step 514 to determine user's access privileges based on the credentials sent by authentication server 208. At step 516, reverse proxy 204 translates any external addresses to intranet addresses in the request. At step 518, the request is processed.

Authentication Process Flow

In an embodiment of the invention, a new user is authenticated to ensure that the user is an authorized user. In addition, an existing user (e.g., a user who was previously authenticated with reverse proxy 204) may be required to undergo a new authentication. For example, a user whose session exceeds a threshold amount of time may be required re-authenticate, or from whom a request has not been received for a threshold period of time. Figures 6A-6B provide an authentication process flow according to an embodiment of the invention.

Step 602 is initiated after a request is received from client 202 and reverse proxy 204 determines that the user needs to be authenticated. At step 602, reverse proxy 204 sends a request to client 202 for the user's information. At step 604, the client's userid is sent to authentication server 208 once it is received from client 202 by reverse proxy 204.

At step 606, reverse proxy 204 receives the challenge generated by authentication server 208. At step 608, reverse proxy 204 obtains a cookie that uniquely identifies the session that the user is establishing between client 202 and reverse proxy 204.

At step 610, reverse proxy 204 retains the cookie, challenge and userid for the user. For example, reverse proxy 204 can send the cookie, challenge and userid values to state server 206 in a storage request. At step 612, reverse proxy 204 sends the challenge to client 202 along with the cookie.

Referring to step 614 of Figure 6B, reverse proxy 204 receives a result and cookie from the client. At step 616, reverse proxy 204 obtains the challenge and userid associated with the cookie sent by client 202. At step 618, reverse proxy 204 sends the challenge, result and userid to authentication server 208. At step 620, reverse proxy 204 receives either a rejection or the user's credentials in response to the challenge, result and userid. At step 624, processing continues at step 510 of Figure 5 to determine whether authentication succeeded or failed.

Request Processing Process Flow

In an embodiment of the invention, a user's request received by reverse proxy 204 is forwarded to the intranet, if the user is an authenticated user and is authorized to access the referenced intranet resources. Figures 5 and 6A-6B include steps for ensuring that a user is an authenticated user. If a request is from an authenticated user, reverse proxy 204 translates external references to intranet references (see step 516 of Figure 5). Figure 7 provides a process flow for processing an authenticated user's request according to an embodiment of the invention.

At step 702, reverse proxy 204 determines whether the user is authorized to access a referenced intranet resource (e.g., by comparing the user's credentials with the reference). If the user does not have authorization to access the referenced intranet resource, processing continues at step 716 to generate an error.

If the user has authorization to access the referenced intranet resource, processing continues at step 704 to include user authentication information in the request and forward the request to the intranet (e.g., proxy server 210). The request is forwarded within the intranet to the referenced resource.

Where the request results in a response, the response is forwarded back to the user. At step 706, a response to the request is received by reverse proxy 204. At step 708, reverse proxy 204 translates intranet references contained in the response to external references. At step 710, the filtered content (i.e., the content containing the external references) is forwarded to client 202. Request processing finishes for the current request at step 712.

Thus, a method and apparatus for authenticated secure access to computer networks has been provided in conjunction with one or more specific embodiments. The invention is defined by the claims and their full scope of equivalents.

CLAIMS

1. In a computer system, a method of accessing a internal computer network from an external source comprising:

receiving a message from an external source at said internal computer network, said message comprising a plurality of external references to a first plurality of resources of said computer network;

translating said plurality of external references to a plurality of internal references for said first plurality of resources;

translating a plurality of responses to said received message to be sent to said external source, said plurality of responses comprising a plurality of internal references to a second plurality of resources of said computer network into a plurality of external references to said second plurality of resources.

2. The method of claim 1 further comprising:

performing an authentication when said received message is received from a user.

3. The method of claim 2 wherein said performing an authentication further comprises:

forwarding a challenge to said user;

validating a result generated by said user from said challenge;

authenticating said user when said result is valid.

4. The method of claim 1 wherein said translating said plurality of external references further comprises:

determining access privileges for a user, said access privileges identifying a set of mappings comprising a plurality of external references mapping entries and a corresponding plurality of internal references mapping entries;

translating those of said plurality of external references for which said user is authorized based on said access privileges.

5. The method of claim 4 wherein said translating said plurality of external references further comprises:

matching said one of said plurality of external references to said plurality of external reference mapping entries;

translating said one of said plurality of external references when said one of said plurality of external references matches at least one of said plurality of external reference mapping entries, said one of said plurality of external references being translated using one of said internal reference mapping entries that corresponds to said at least one of said plurality of external reference mapping entries.

6. The method of claim 1 wherein said computer network comprises a plurality of applications, said method further comprising:

obtaining authentication information when a user attempts to access said computer network from an external source;

forwarding said authentication information to each of said plurality of applications as needed as said user attempts to access said each of said plurality of applications.

7. The method of claim 1 further comprising:

determining whether a cookie is transmitted with an attempt by a user to access said computer network;

determining whether said cookie is valid, if said cookie is transmitted;

authenticating said user, if at least one of said cookie not being transmitted and said cookie is not valid conditions occurs.

8. A system comprising:

an external communication network comprising a plurality of computing devices;

a reverse proxy coupled to said external communication network;

an internal communications network coupled to said reverse proxy;

a set of mappings coupled to said reverse proxy, said set of mappings configured to map between an external reference and an internal reference to a resource of said internal communications network.

9. The system of claim 8 wherein said reverse proxy is configured to translate from said external reference to said internal reference when said external reference is received from said external communications network.

10. The system of claim 8 wherein said reverse proxy is configured to translate from said internal reference to said external reference when said internal reference is sent from said internal communications network to said external communications network.

11. The system of claim 8 further comprising an authentication server, said authentication coupled to said reverse proxy and configured to authenticate a user on said external communications network attempting to access said internal communications network.

12. The system of claim 8 wherein said reverse proxy is coupled to said external communications network via a plurality of intermediate servers.

13. The system of claim 8 wherein said internal communications network further comprises a plurality of application servers, said plurality of application servers are coupled to reverse proxy via a plurality of proxy servers.

14. A computer program product comprising:

a computer usable medium having computer readable program code embodied therein for accessing a internal computer network from an external source comprising:

computer readable program code configured to cause a computer to receive a message from an external source at said internal computer network, said message comprising a plurality of external references to a first plurality of resources of said computer network;

computer readable program code configured to cause a computer to translate said plurality of external references to a plurality of internal references for said first plurality of resources;

computer readable program code configured to cause a computer to translate a plurality of responses to said received message to be sent to said external source, said plurality of responses comprising a plurality of internal references to a second plurality of resources of said computer network into a plurality of external references to said second plurality of resources.

15. The computer program product of claim 14 further comprising:

computer readable program code configured to cause a computer to perform an authentication when said received message is received from a user.

16. The computer program product of claim 15 wherein said computer readable program code configured to cause a computer to perform an authentication further comprises:

computer readable program code configured to cause a computer to forward a challenge to said user;

computer readable program code configured to cause a computer to validate a result generated by said user from said challenge;

computer readable program code configured to cause a computer to authenticate said user when said result is valid.

17. The computer program product of claim 14 wherein said computer readable program code configured to cause a computer to translate said plurality of external references further comprises:

computer readable program code configured to cause a computer to determine access privileges for a user, said access privileges identifying a set of mappings comprising a plurality of external references mapping entries and a corresponding plurality of internal references mapping entries;

computer readable program code configured to cause a computer to translate those of said plurality of external references for which said user is authorized based on said access privileges.

18. The computer program product of claim 17 wherein said computer readable program code configured to cause a computer to translate said plurality of external references further comprises:

computer readable program code configured to cause a computer to match said one of said plurality of external references to said plurality of external reference mapping entries;

computer readable program code configured to cause a computer to translate said one of said plurality of external references when said one of said plurality of external references matches at least one of said plurality of external reference mapping entries, said one of said plurality of external references being translated using one of said internal reference mapping entries that corresponds to said at least one of said plurality of external reference mapping entries.

19. The computer program product of claim 14 wherein said computer network comprises a plurality of applications, said computer program product further comprising:

computer readable program code configured to cause a computer to obtain authentication information when a user attempts to access said computer network from an external source;

computer readable program code configured to cause a computer to forward said authentication information to each of said plurality of applications as needed as said user attempts to access said each of said plurality of applications.

20. The computer program product of claim 14 further comprising:

computer readable program code configured to cause a computer to determine whether a cookie is transmitted with an attempt by a user to access said computer network;

computer readable program code configured to cause a computer to determine whether said cookie is valid, if said cookie is transmitted;

computer readable program code configured to cause a computer to authenticate said user, if at least one of said cookie not being transmitted and said cookie is not valid conditions occurs.

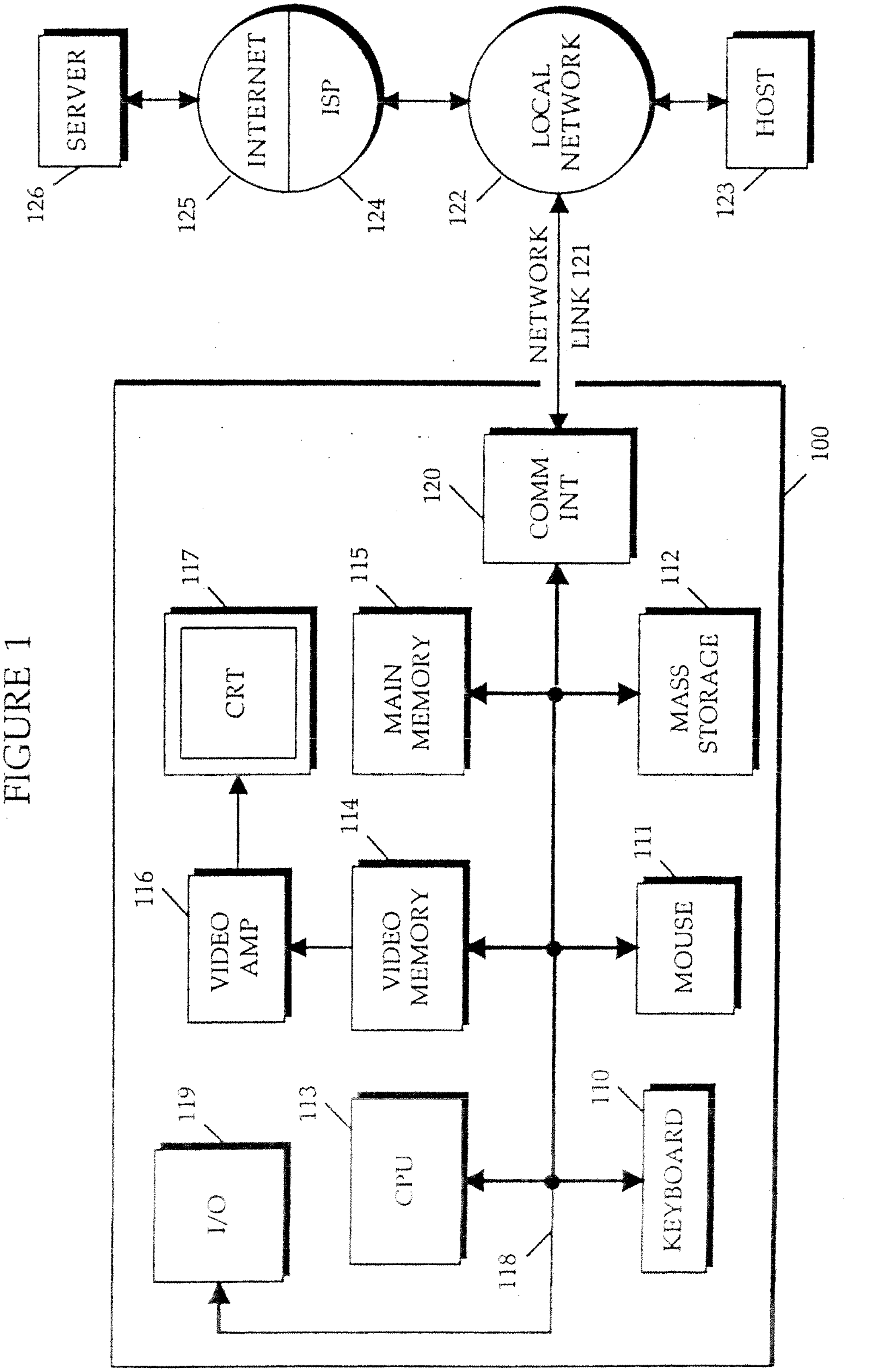

SERVER
126
INTERNET
ISP
125
124
LOCAL NETWORK
122
HOST
123
NETWORK LINK 121
100
COMM INT
120
CRT
117
MAIN MEMORY
115
MASS STORAGE
112
VIDEO AMP
116
VIDEO MEMORY
114
MOUSE
111
I/O
119
CPU
113
KEYBOARD
110
118

FIGURE 1

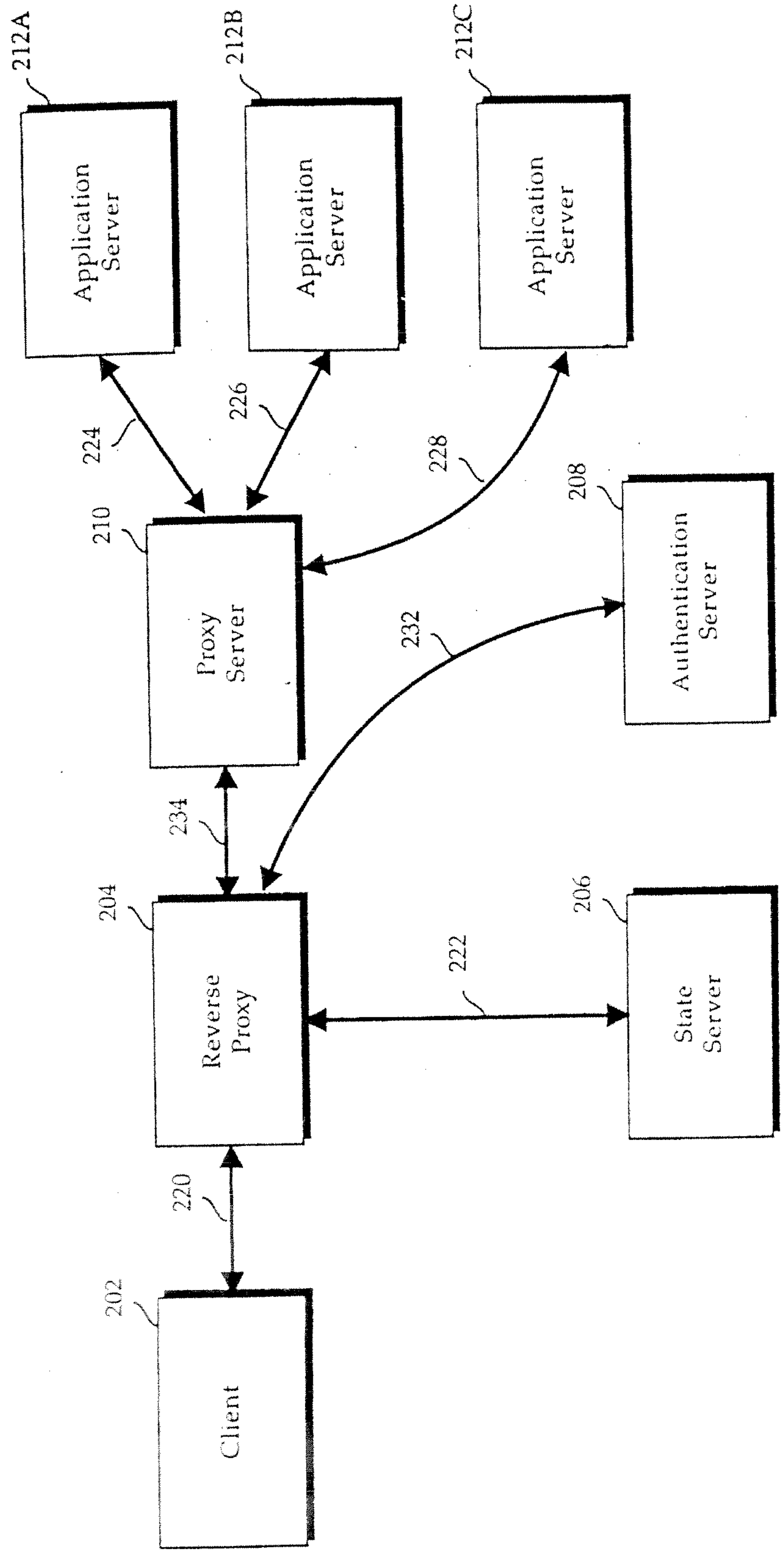

212A
Application Server
212B
Application Server
212C
Application Server
224
226
228
210
Proxy Server
208
Authentication Server
232
234
204
Reverse Proxy
222
206
State Server
220
202
Client

Figure 2A

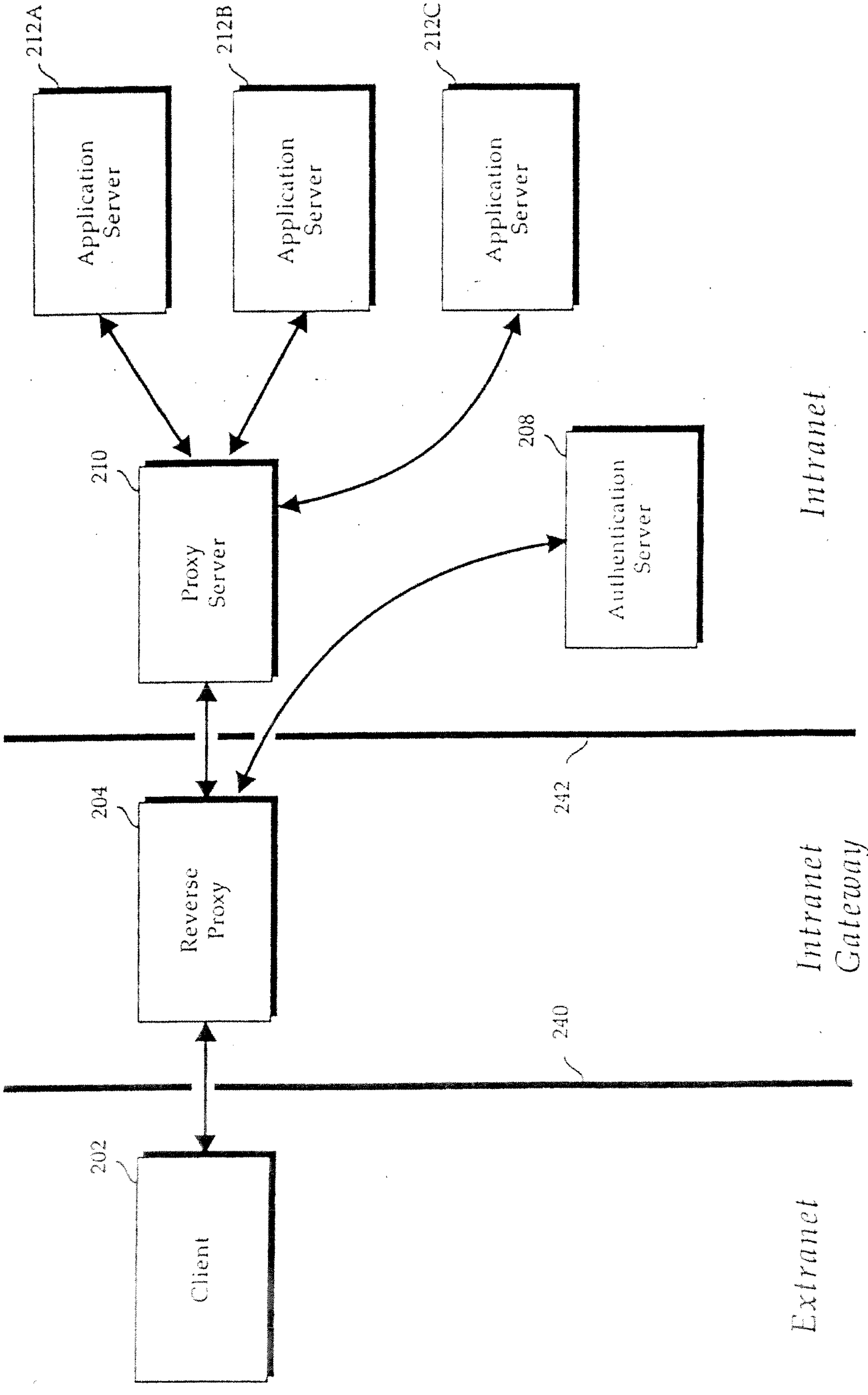

212A
Application Server
212B
Application Server
212C
Application Server
210
Proxy Server
208
Authentication Server
Intranet
248
242
204
Reverse Proxy
Intranet Gateway
246
240
202
Client
Extranet
244

Figure 2B

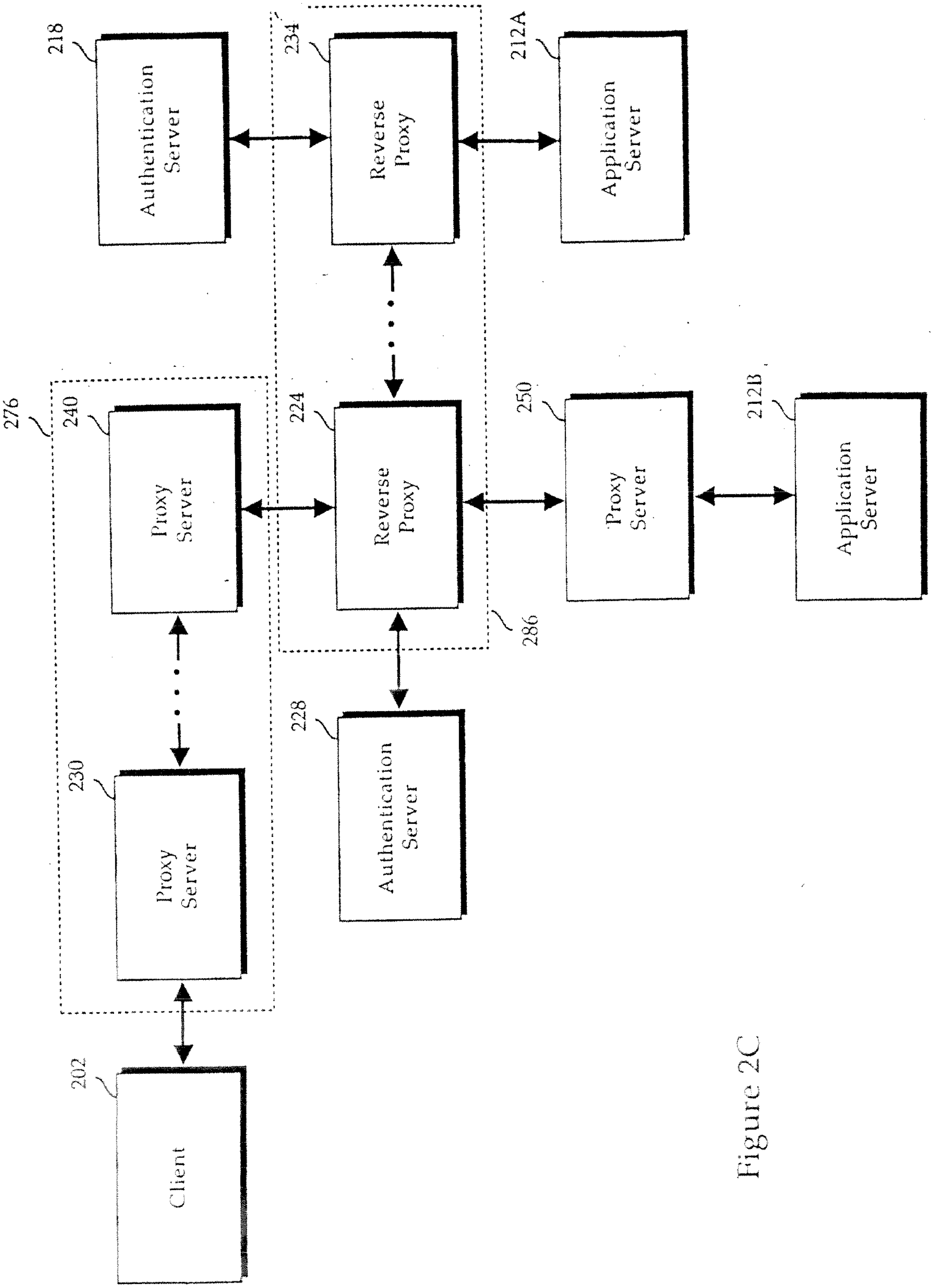

Client
202
Proxy Server
230
Proxy Server
240
276
Reverse Proxy
224
286
Authentication Server
228
Reverse Proxy
234
Authentication Server
218
Proxy Server
250
Application Server
212A
Application Server
212B

Figure 2C

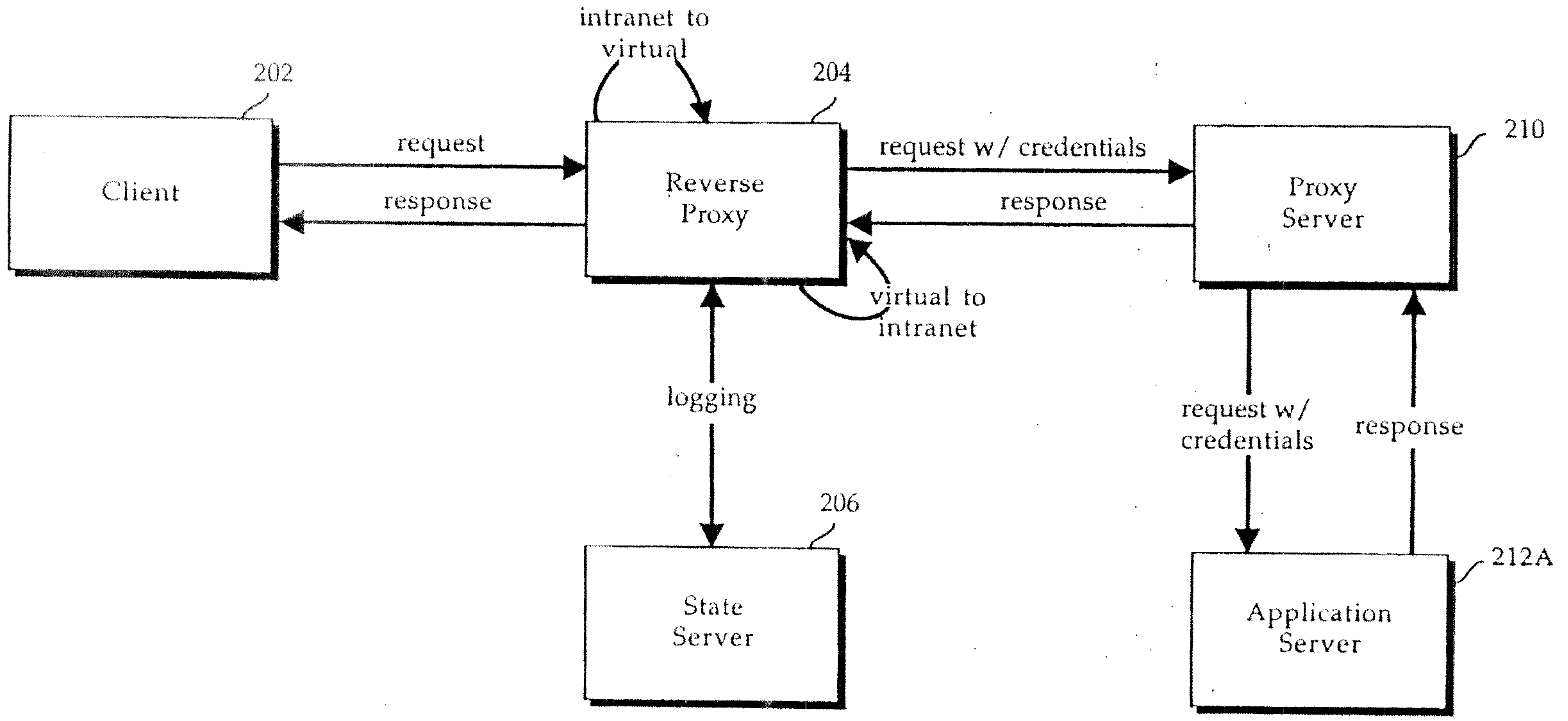

intranet to
virtual
202
204
Client
request
response
Reverse
Proxy
request w/ credentials
response
Proxy
Server
210
virtual to
intranet
logging
request w/
credentials
response
206
State
Server
212A
Application
Server

Figure 3

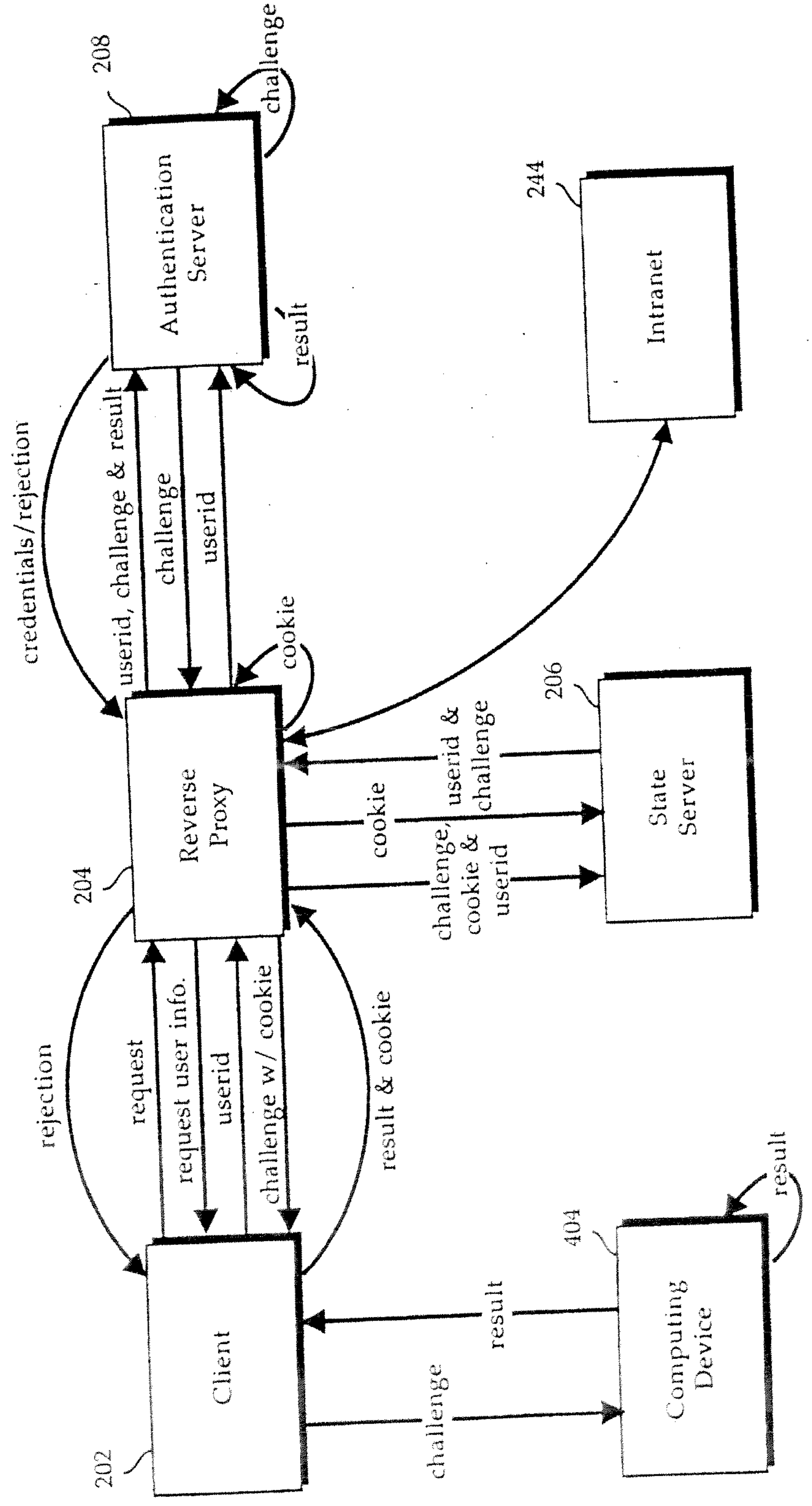
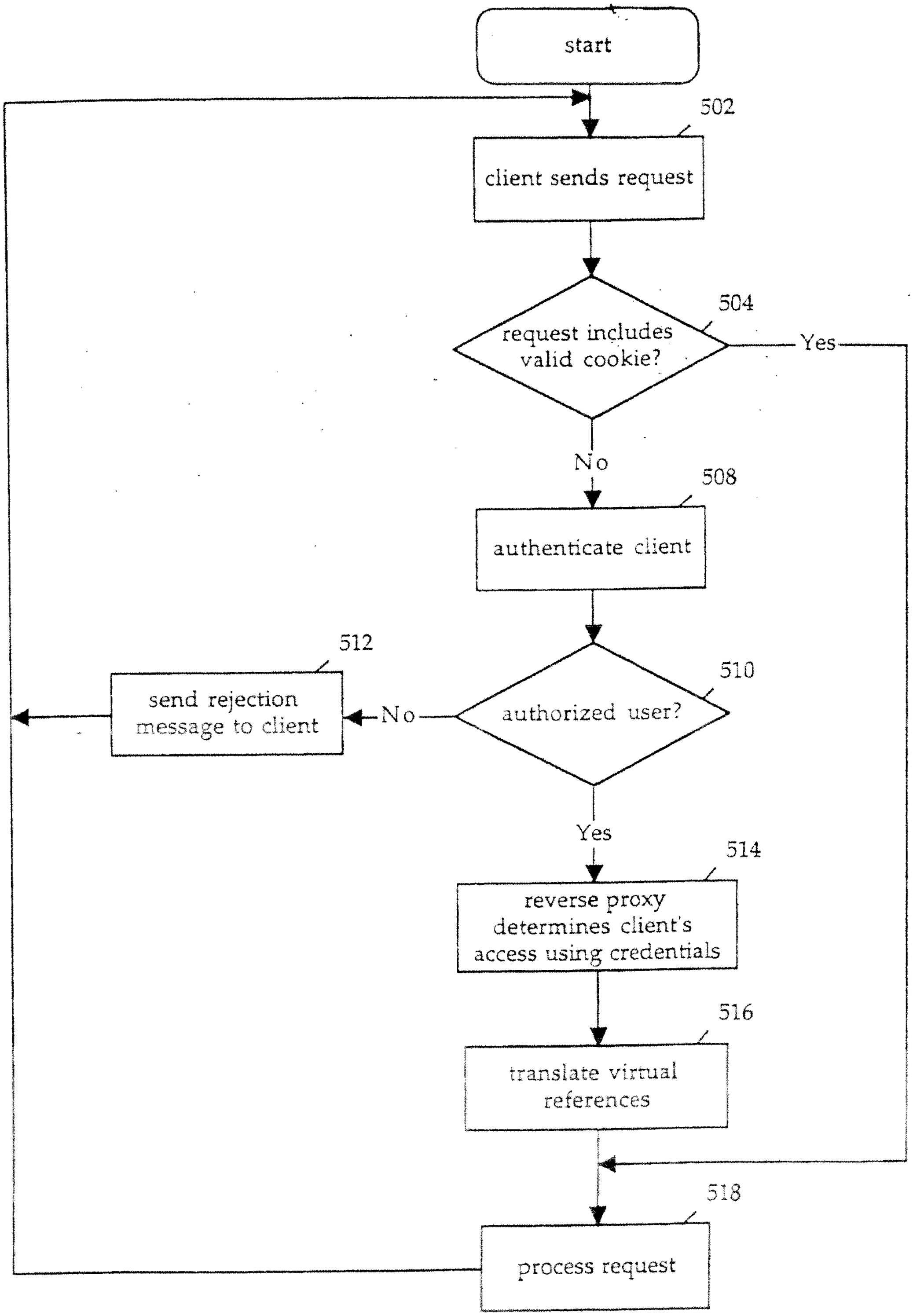

208
Authentication Server
challenge
result
244
Intranet
credentials/rejection
userid, challenge & result
challenge
userid
cookie
204
Reverse Proxy
206
State Server
userid & challenge
cookie
challenge, cookie & userid
rejection
request
request user info.
userid
challenge w/ cookie
result & cookie
202
Client
404
Computing Device
result
challenge
result

Figure 4



start
502
client sends request
504
request includes valid cookie?
Yes
No
508
authenticate client
512
send rejection message to client
No
510
authorized user?
Yes
514
reverse proxy determines client's access using credentials
516
translate virtual references
518
process request

Figure 5

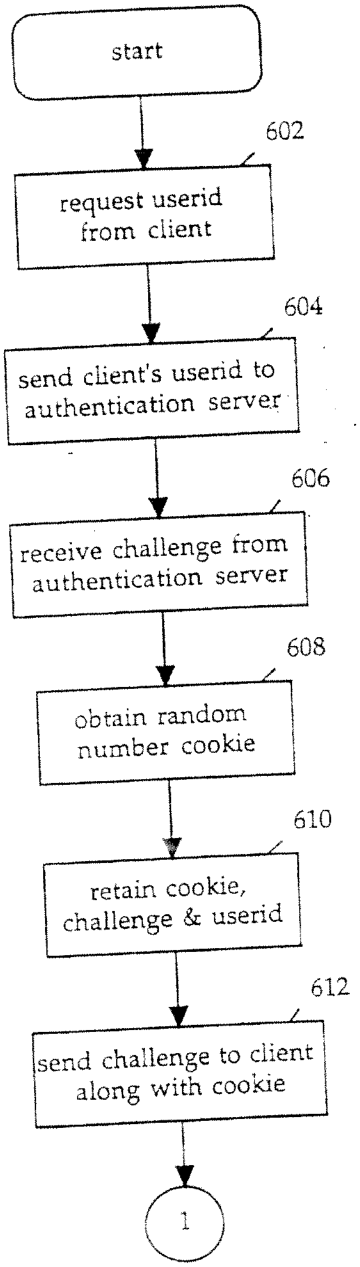

start
602
request userid from client
604
send client's userid to authentication server
606
receive challenge from authentication server
608
obtain random number cookie
610
retain cookie, challenge & userid
612
send challenge to client along with cookie
1

Figure 6A

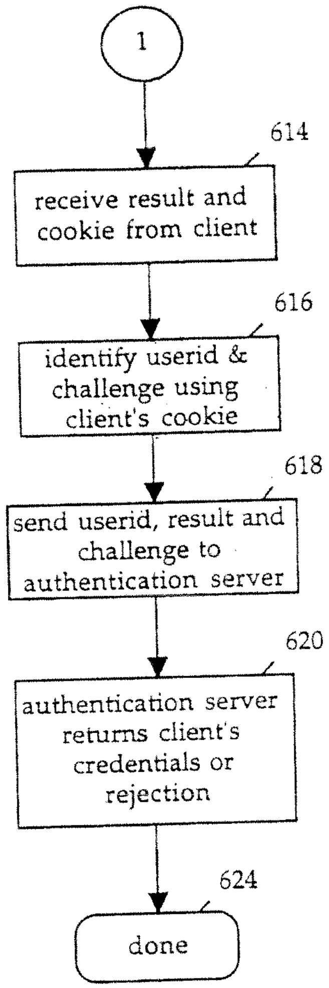

1
614
receive result and cookie from client
616
identify userid & challenge using client's cookie
618
send userid, result and challenge to authentication server
620
authentication server returns client's credentials or rejection
624
done

Figure 6B

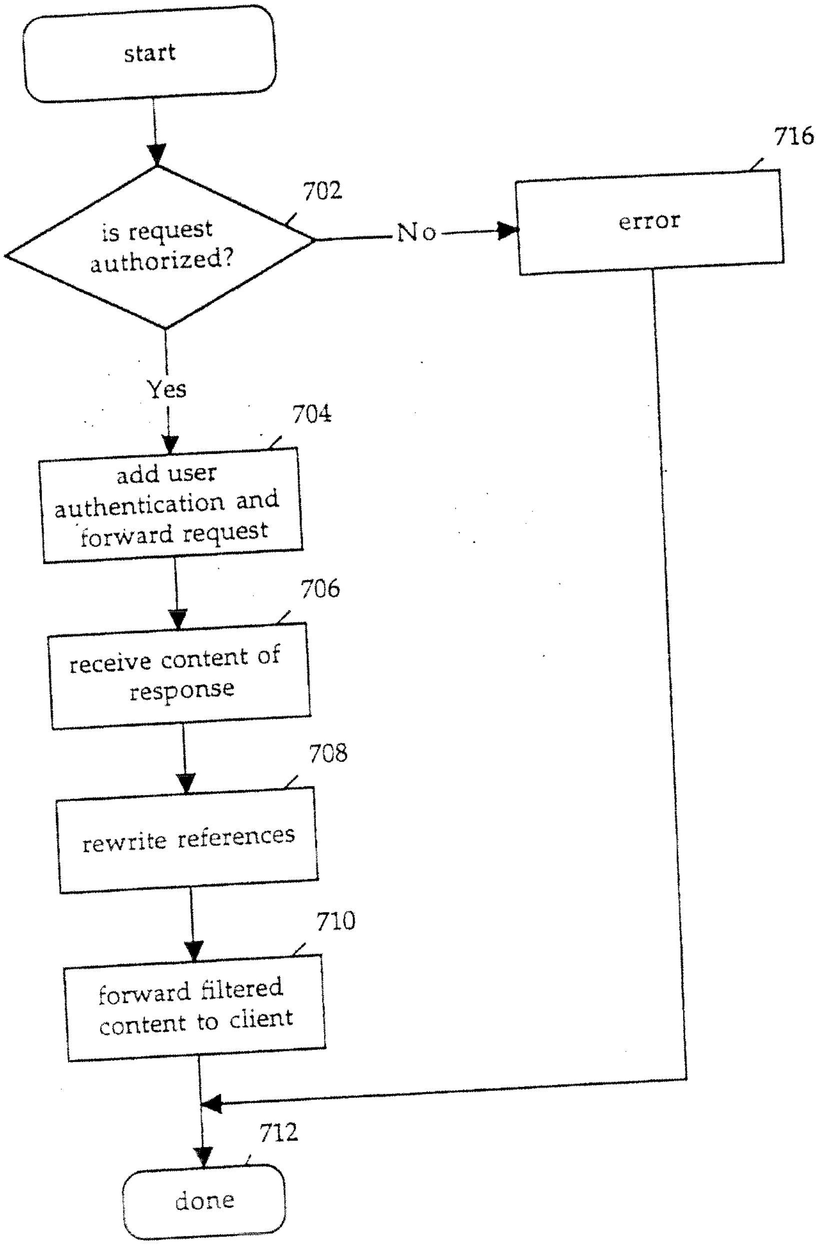

start
is request authorized?
702
No
error
716
Yes
add user authentication and forward request
704
receive content of response
706
rewrite references
708
forward filtered content to client
710
712
done

Figure 7

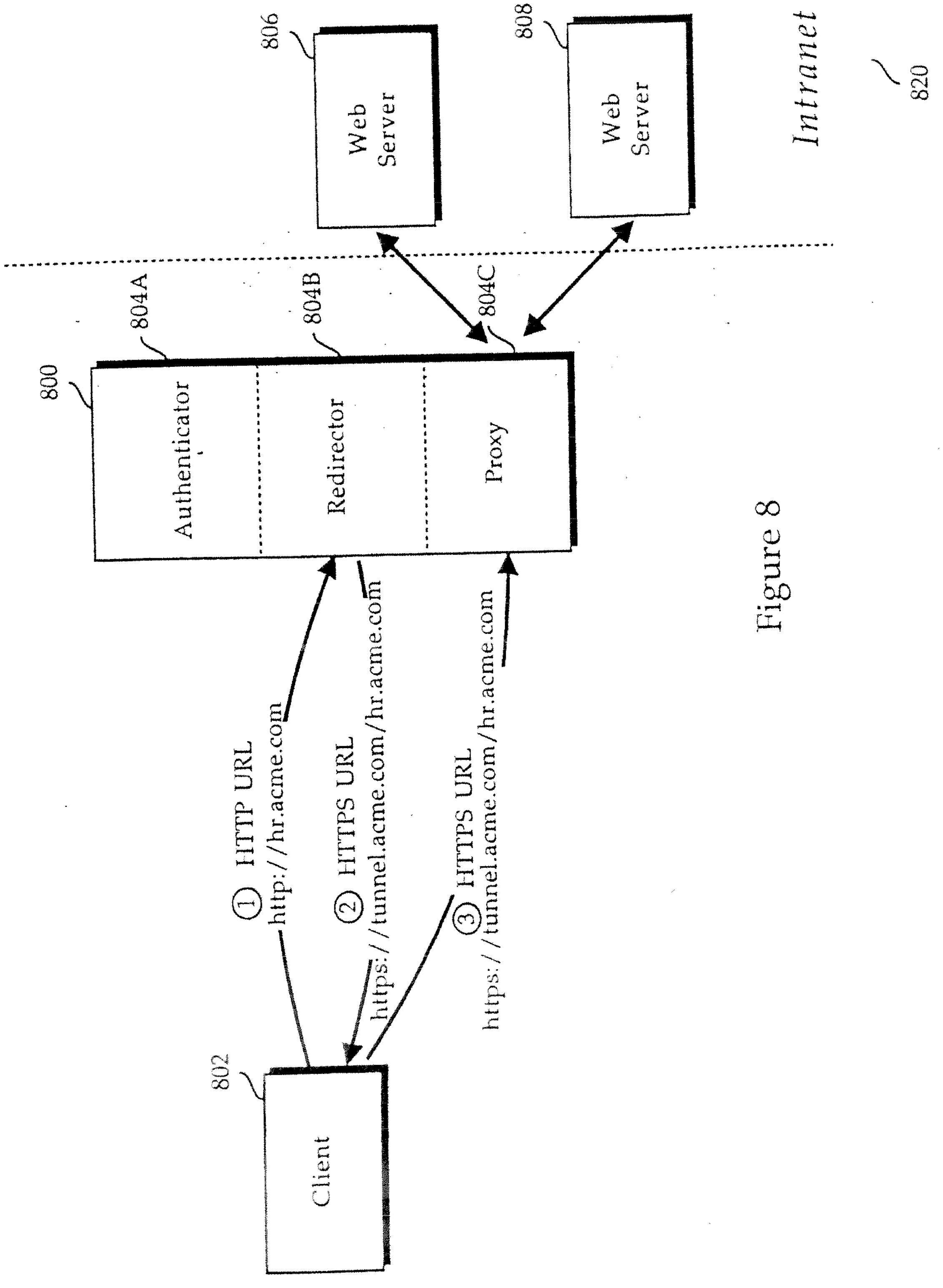

806
Web Server
808
Web Server
Intranet
820
800
804A
804B
804C
Authenticator
Redirector
Proxy
① HTTP URL
http://hr.acme.com
② HTTPS URL
https://tunnel.acme.com/hr.acme.com
③ HTTPS URL
https://tunnel.acme.com/hr.acme.com
802
Client

Figure 8